# A Two-stage Information Filtering Based on Rough Decision Rule and Pattern Mining

Xujuan Zhou*, Yuefeng Li*, Peter Bruza*, Yue Xu* and Raymond Lau
Faculty of Science and Technology, Queensland University of Technology, Brisbane, Australia*
Department of Information Systems, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong
Email: {x.zhou, y2.li, p.bruza yue.xu}@qut.edu.au, raylau@cityu.edu.hk

***Abstract*— Information Overload and Mismatch are two fundamental problems affecting the effectiveness of information filtering systems. Even though both term-based and pattern-based approaches have been proposed to address the problems of overload and mismatch, neither of these approaches alone can provide a satisfactory solution to address these problems. This paper presents a novel two-stage information filtering model which combines the merits of term-based and pattern-based approaches to effectively filter sheer volume of information. In particular, the first filtering stage is supported by a novel rough analysis model which efficiently removes a large number of irrelevant documents, thereby addressing the overload problem. The second filtering stage is empowered by a semantically rich pattern taxonomy mining model which effectively fetches incoming documents according to the specific information needs of a user, thereby addressing the mismatch problem. The experimental results based on the RCV1 corpus show that the proposed two-stage filtering model significantly outperforms the both term-based and pattern-based information filtering models.**

***Index Terms*— Information Filtering, User Profiles, Rough Set Theory, Pattern Mining**

## I. Introduction

An Information Filtering (IF) [1] system monitors an incoming document stream to find the documents that match information needs of users. With information filtering, the representation of the user information needs is variously referred to as user profiles or a topic profile where the filters are applied to the dynamic streams of incoming data. Unlike the traditional search query, user profiles are persistent, and tend to reflect a long-term information need [1].

The traditional IF systems make the decision of rejection or reception for a document when it arrives in the stream. The relevant document is displayed to its users without further scrutiny. This decision-making is completed in one step. Such systems often have difficulties in dealing with issues, such as the feature selection on how to remove noisy and non-relevant features, and threshold setting (how to learn the optimal threshold). To improve the robustness of the IF systems, this paper will propose a novel two-stage framework for information filtering.

To illustrate the two-stage IF model, consider an example that may occur in a TV series. Louisa is a girl from a big city looking for a partner. There are three strategies Louisa may adopt, (i) set herself criteria, check out the information of as many people as possible. When Mr Right meets the criteria, she chooses him and lives happily ever after, (ii) date with everyone that is available, rank them according to suitability, then choose the highest-ranked person and live happily ever after, (iii) set herself criteria for rejection, date with a small number of people then choose the best fit.

The first approach has some obvious setbacks. If the criteria are set too high, Louisa may never find Mr Right. If the criteria are too low, she would have difficulties choosing Mr Right. The second strategy is also not practical. What is the point in wasting the energy with every one and ranking the obviously not suitable people? The third approach is a two-stage method. It is the only sensible and efficient way in the three approaches.

Within the new two-stage IF framework, the first filtering stage is supported by a novel rough analysis model which efficiently removes a large number of irrelevant documents. The intention after the first stage is that only a relatively small amount of potentially highly relevant documents remain as the input to the second stage. The second filtering stage is empowered by a semantically rich pattern taxonomy mining model which effectively rank incoming documents according to the specific information needs of a user and fetches the top ranking documents for a user. Our experimental results confirm that the proposed two-stage IF model which combines the advantages of term-based and pattern-based filtering performs significantly better than other state-of-the-art IF models.

The remainder of the paper is organized as follows. Section 2 highlights previous research in related areas. Section 3 introduces the Rough Set Decision Rule-based Topic Filtering. Section 4 presents filtering model based on the pattern taxonomy mining. The empirical results are reported in Section 5. Section 6 describes the findings of the experiments and discusses the results. Concluding remarks are sketched in Section 7.

## II. Related Work

### A. Term-based IF

IF systems were originally considered to have the same function as IR systems did. Different from IR systems, IF systems were commonly personalized to support long-term information needs of users [1]. The main distinction


doi:10.4304/jetwi.2.4.326-332

between IR and IF was that IR systems used "queries" but IF systems used "user profiles".

The representation of the user information need is variously refereed to as user profiles, or topic profiles. As the quality of the profiles directly influences the quality of information filtering, the issue of how to built accurate, reliable profiles is a crucial concern [2]. The tasks of the filtering track in TREC included batch and routing filtering, and adaptive filtering [3]. A batch filtering system uses a retrieval algorithm to score each incoming document. If the score is greater than a specified threshold, then the document is delivered to the user. The routing filtering systems are more similar to the retrieval systems, the profile remains constant and the task is to match an incoming stream of documents to a set of profiles. Both systems need to return a ranked list of documents. Adaptive filtering involves feedback to dynamically adapt IF systems [**?**]. The profile is adapted dynamically in the presence of feedback.

The term-based IF systems used terms to represent the user profiles. Such profiles are the most simplest and common representation of the profiles. For examples: the probabilistic models [4], BM25 [5], rough set-base models [6], [7], and ranking SVM [8] based filtering models used the term-based user profiles. The advantage of term-based model is efficient computational performance as well as mature theories for term weighting, which have emerged over the last couple of decades from the IR and machine learning communities. However, term-based models suffer from the problems, such as, the relationship among the words cannot be reflected [8] and also, only considering single words as features is the semantic ambiguity. For example: the synonym problem is a word that shares the same meaning as another word (for example, "taxi" and "cab"), and the homonym problem is a word that is pronounced, and sometimes spelled, in the same way as another word but has a different meaning (for example, "there" and "their").

Phrase-based method is therefore proposed. This method used the multiple words (phrases) as features to solve the semantic ambiguity problem. It is believed that the simple term-based representation of the profile is usually inadequate, because single words are rarely sufficiently specific for accurate discrimination. However, Fuhr [9] investigated the probabilistic models in IR and pointed out that a dependent model for phrases is not sufficient, because only the occurrence of the phrase components in a document is considered, but not the syntactical structure of the phrases. Moreover, the certainty of identification should also be regarded, such as, whether the words occur adjacent or only within the same paragraph.

### *B. Pattern Mining for IF*

Pattern mining has been extensively studied in data mining communities for many years. A variety of efficient algorithms such as Apriori-like algorithms [10], PrefixSpan [11], FP-tree [12], SPADE [13], SLPMiner [14] and GST [15] have been proposed.

In the field of text mining, pattern mining techniques can be used to find various text patterns, such as co-occurring terms and multiple grams, maximal frequent patterns, and closed patterns, for building up a representation with these new types of features. In [16], data mining techniques have been used for text analysis by extracting co-occurring terms as descriptive phrases from document collections. However, the effectiveness of the text mining systems using phrases as text representation showed no significant improvement. Mining maximal frequent patterns [**?**] was also proposed to reduce the time complexity of mining all frequent patterns, where an itemset (or a pattern) was maximal frequent if it had no superset that was frequent. The similar idea, maximal association rules, was also used for text mining [**?**], [**?**], where users provided categories for finding maximal rules they wanted.

Maximal association mining ignored all of small patterns. However, some small patterns can be very useful. The notion of closed patterns has its origins in the mathematical theory of Formal Concept Analysis introduced in [**?**]. Closed patterns were used to prune some smaller useless patterns [**?**] and that have been used for improve the effectiveness of text mining [17].

Typically, text mining discusses associations between terms at a broad spectrum level, paying little heed to duplications of terms, and labeled information in the training set [**?**]. Usually, the existing data mining techniques return numerous discovered patterns (e.g., sets of terms) from a training set. Not surprisingly, among these patterns, there are many redundant patterns [18]. Nevertheless, the challenging issue is how to effectively deal with the large amount of discovered patterns.

Sequential closed patterns used in data mining community have turned out to be a promising alternative to phrases [19]. To consider the very important semantic relationships between the terms, a pattern taxonomy model (PTM) for IF has been proposed in [20]. Pattern taxonomy is a tree-like hierarchy that reserves the sub-sequence (that is, "is-a") relationship between the discovered sequential patterns. These pattern based approaches have shown encouraging improvements on effectiveness, but at the expense of computational efficiency. In regard to the aforementioned problem of redundancy and noise, PTM adopts the concept of closed patterns, or pruned non-closed patterns. However, it is a still challenging issue for PTM to deal with low frequency patterns because the measures used in data mining to learn profiles turn out be not suitable in the filtering stage. By way of illustration, given a specified topic, a highly frequent pattern is usually a general pattern, or a specific pattern of low frequency. This parallels the situation in term indexing where words of high frequency (stop words) or very low frequency (highly information bearing uncommon words) are not considered useful.

The two-stage IF model was initially proposed in [**?**] and further developed in this paper. The extensive experi-

ments have been conducted to verify the threshold setting method in the first stage. The paper will demonstrate that exploit rough set-base reasoning and pattern mining approaches to develop two-stage IF system can achieve the better filtering performance.

## III. Rough Set Decision Rule-based Topic Filtering

Inevitably, IF systems based on user profiles have to deal with the uncertainties of the users' information needs. Even if the perfect formulation of the user profiles is achievable by machine-learning, the information filter would never be faultless with the system acting alone. Most often, the users themselves are not certain of what they are looking for. To deal with the uncertainty issues, a Rough Set-based IF model(RSIF) has been developed in [21]. Based on the rough set theory [22], the decision rules for the partitioning of the incoming document stream into the positive, boundary and negative regions have been developed in this model. There are two key tasks in developing a RSIF model. The first one is using discovered rough patterns to represent the topic profiles. The second task is deciding an optimal threshold based on the obtained topic profiles.

### A. Topic Profiles

In this section, we first discuss how to represent positive documents in term weight distributions. We also describe the method for deciding suitable thresholds for filtering out likely non-relevant documents for the second stage. In addition, we present topic filtering algorithms in this section.

Let $D$ be a training set of documents, which consists of a set of positive documents, $D^+$; and a set of negative documents, $D^-$. Let $T = \{t_1, t_2, \ldots, t_m\}$ be a set of terms (or keywords) which are extracted from the set of positive documents, $D^+$.

A set of terms is referred to as a *termset*. Given a positive document $d$ and a term $t$, $tf(d,t)$ is defined as the number of occurrences of $t$ in $d$. A set of term frequency pairs, $p_d = \{(t,f)|t \in T, f = tf(t,d) > 0\}$, is referred to as an initial *r-pattern* (rough pattern) in this paper.

Let $termset(p) = \{t|(t,f) \in p\}$ be the *termset* of $p$. In this paper, r-pattern $p_1$ equals to r-pattern $p_2$ if and only if $termset(p_1) = termset(p_2)$. A r-pattern is uniquely determined by its *termset*. Two initial r-patterns can be composed if they have the same *termset*. In this paper, we use the composition operation, $\oplus$, that defined in **[?]** to compose r-patterns. For example,

$$\{(t_1,2),(t_2,5)\} \oplus \{(t_1,1),(t_2,3)\} = \{(t_1,3),(t_2,8)\}$$

(Notice: $\oplus$ is also suitable for patterns with different termsets, e.g., $\{(t_1,2),(t_2,5)\} \oplus \{(t_1,1)\} = \{(t_1,3),(t_2,5)\}$).

Based on the above definitions, we can obtain a set of composed r-patterns in $D^+$, $RP = \{p_1, p_2, \ldots, p_r\}$, where $r \leq n$, where $n = |D^+|$ is the number of positive documents in $D$. The support of a r-pattern $p_i$ is the fraction of the initial r-patterns that are composed to form $p_i$.

TABLE I.
A SET OF POSITIVE DOCUMENTS

| *Document* | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ |
|---|---|---|---|---|---|---|---|
| $d_1$ | 2 | 1 | 0 | 0 | 0 | 0 | 0 |
| $d_2$ | 0 | 0 | 2 | 1 | 0 | 1 | 0 |
| $d_3$ | 0 | 0 | 3 | 1 | 1 | 1 | 0 |
| $d_4$ | 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| $d_5$ | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| $d_6$ | 2 | 1 | 0 | 0 | 0 | 1 | 1 |

TABLE II.
A SET OF DISCOVERED R-PATTERNS

| *R-pattern* | $t_1$ | $t_2$ | $t_3$ | $t_4$ | $t_5$ | $t_6$ | $t_7$ | *support* |
|---|---|---|---|---|---|---|---|---|
| $p_1$ | 2 | 1 | 0 | 0 | 0 | 0 | 0 | 1/6 |
| $p_2$ | 0 | 0 | 2 | 1 | 0 | 1 | 0 | 1/6 |
| $p_3$ | 0 | 0 | 4 | 2 | 2 | 2 | 0 | 1/3 |
| $p_4$ | 3 | 2 | 0 | 0 | 0 | 2 | 2 | 1/3 |

Table I shows a set of positive documents in a training set, where $T = \{t_1, t_2, \ldots, t_7\}$, and the numbers are term frequencies in the corresponding documents. We also can view the documents in Table I initial r-patterns.

Table II illustrates the discovered r-patterns and their supports by using the composition operation to the initial r-patterns in Table I, where $p_1 = d_1$, $p_2 = d_2$, $p_3 = d_3 \oplus d_4$, and $p_4 = d_5 \oplus d_6$ for all discovered r-patterns.

Formally, the relationship between the r-patterns and the terms can be described as the following *association mapping*, if term frequencies are considered:

$$\beta : RP \rightarrow 2^{T \times [0,1]}, \tag{1}$$

such that

$$\beta(p_i) = \{(t_1,w_1),(t_2,w_2),\ldots,(t_k,w_k)\},$$

where $p_i \in RP$ is an r-pattern; and weight $w_i$ is:

$$w_i = \frac{f_i}{\sum_{j=1}^{k} f_j}$$

if it is assumed

$$p_i = \{(t_1,f_1),(t_2,f_2),\ldots,(t_k,f_k)\}.$$

$\beta(p_i)$ is called the *normal form* of the r-pattern $p_i$ in this thesis. The association mapping $\beta$ can derive a probability function for the weight distribution of terms on $T$ to show the importance of the terms in the positive documents, which satisfies:

$$pr_\beta(t) = \sum_{p_i \in RP, (t,w) \in \beta(p_i)} support(p_i) \times w \tag{2}$$

for all $t \in T$.

Based on the above discussion, a positive document $d_i$ can be described as an event that represents what the users want with the probability value

$$prob(d_i) = \sum_{t \in d \cap T} pr_\beta(t).$$

As shown above, the r-patterns can be extracted from the positive documents in the training set. The discovered r-patterns represent the topic profiles. In the proposed two-stage IF system, the topic profiles are employed to filter out the most of the irrelevant documents rather than for identifying the relevant documents. The main objective of the first stage of the proposed system is to reduce the "noises".

### B. Optimal Rough Threshold

To work out the suitable thresholds, it is assumed that document $d$ is irrelevant if it is not closed to the common feature of the topic profiles in the training set. For a given topic, it consists of a set of the positive document, $D^+$. Each document $d_i$ in $D^+$ is represented by $prob(d_i) = \sum_{t \in d \cap T} pr_\beta(t)$. This means that each document $d_i$ has a weight $W_d = \sum_{t \in d \cap T} pr_\beta(t)$. To capture the common feature of the topic from the training data, the distributions of the document weights for a given topic must be first understood.

Only positive documents are used for simulating the user profiles. After obtaining the documents weight using the training set data, a normal distribution is employed to represent the distributions of the weights of the documents. For a given topic, a normal distribution curve would have fit the distribution of the document weights. Statistical features, such as the mean, variance and skewness, can be found from the best-fit probability functions. These statistical features can be used to decide the thresholds. The thresholds determined by the statistical features would capture not only the meaning of the relevance, but also the certainty of the relevance of the user profiles. Therefore, the common feature of a topic/user profile will be better represented by this proposal. According to the statistical approach, the common feature, $\xi_j$ for a topic can be modeled as:

$$\xi_j = \frac{1}{n} \sum_{d_i \in D^+} prob(d_i);$$

where $n$ is the number of the positive documents, $n = |D^+|$. In fact, $\xi_j$ is the mean, $\overline{m}$, of the probabilities of the positive documents in $D^+$. The thresholds, therefore, can be simply determined as $threshold = \xi_j$.

It is reasonable to assume that the weights of the document follow a normally distributed pattern. Many simplistic models assume normal distribution, that is, the data is symmetric about the mean. The normal distribution has a skewness of zero. Using the mean of the Rough Set weights as a threshold would be a good initial choice, because the mean represents the "common feature". However, real data points are not always perfectly symmetric. Skewness is a measure of the asymmetry of the probability distribution of a real-valued random variable.

By observations, the discrimination power of using mean as the threshold is still not high enough. Many non-relevant documents can not be filtered out. The distributions of the weights of the documents have exhibited a high degree of skewness. To obtain the "real" common feature, both the standard derivation and the skewness must be taken into consideration for modeling the document weights. The following features have been used to characterize a histogram in this paper.

$\sigma$ is the standard deviation of the probabilities of positive documents. It is given by:

$$\sigma = \sqrt{\frac{1}{n} \sum_{d_i \in D^+} (prob(d_i) - \overline{m})^2} \tag{3}$$

$\mu$ is the skewness of the probabilities. The skewness is given by:

$$\mu = \frac{\sqrt{n} \sum_{d_i \in D^+} (prob(d_i) - \overline{m})^3}{(\sum_{d_i \in D^+} (prob(d_i) - \overline{m})^2)^{\frac{3}{2}}} \tag{4}$$

Skewness $\mu$ is a measure of the asymmetry degree of a histogram around the mean value. The more asymmetric the distribution, the larger the skewness value. A linear discriminated function is used to make a decision based on features obtained from the above analysis. Therefore, the threshold can be determined as follows:

$$threshold = \xi_j + \gamma(\sigma + \mu) \tag{5}$$

where $\gamma$ is an experimental coefficient.

**Algorithm** $TF1T$ $(D^+, T)$

**Input:** a set of positive documents $D^+$ and a set of terms $T$.
**Output:** a probability function $pr_\beta$ on $T$ and a set of r-patterns $RP$.
**Method:**
(1) $RP = \emptyset$;
(2) **for** (document $d \in D^+$) {
  **for** ($t_i \in T$) let $f_i$ be its term frequency in $d$;
  $\hat{d} = \{(t_1, f_1), \ldots, (t_{|T|}, f_{|T|})\}$;
  $support(\hat{d}) = \frac{1}{|D^+|}$;
  $RP = RP \cup \{\hat{d}\}$ };
(3) $RP = \oplus(RP)$;
(4) **for** (term $t \in T$) $pr_\beta(t) = 0$;
(5) **for** (r-pattern $p \in RP$)
  **for** ($(t, w) \in \beta(p)$)
    $pr_\beta(t) = pr_\beta(t) + w \times support(p)$;

### C. Topic Filtering Algorithms

An efficient training procedure for calculating the derived probability function $pr_\beta$ in topic filtering is described in Algorithm TF1T.

In order to improve efficiency, composition operations are not actually used in Algorithm TF1T. All initial r-patterns $RP$ are collected in steps (1) and (2). The probability distribution over $T$ is then initialized to zero in step (3). Finally, each initial r-pattern is normalized and the probability values are accumulated in step (4). The time complexity of Algorithm TF1T in the training phase is $O(nmq)$, since it only needs a single traversal through positive documents, where $q$ is the average size of documents; $n = |D^+|$ and $m = |T|$.

Algorithm TF1F describes the filtering process of the first stage of the filtering process using the Eq. 5 as the threshold. Furthermore, a relevance value for each document in the testing set is assigned, where $\tau(t, d) = 1$ if $t \in d$; otherwise $\tau(t, d) = 0$.

**Algorithm** $TF1F$ $(D^+, T, pr_\beta, prob, RP, \gamma)$
**Input:** Positive documents $D^+$, terms $T$, $pr_\beta$, $prob$, r-patterns $RP$, and a coefficient $\gamma$.
**Output:** a set of retained (possible relevant) documents $rel$.
**Method:**
(1) $n = \frac{1}{|D^+|}$,
$\overline{m} = \frac{1}{n}\sum_{d_i \in D^+} prob(d_i)$,
$\sigma = \sqrt{\frac{1}{n}\sum_{d_i \in D^+} (prob(d_i) - \overline{m})^2}$,
$skew = \frac{\sqrt{n}\sum_{d_i \in D^+} (prob(d_i) - \overline{m})^3}{(\sum_{d_i \in D^+} (prob(d_i) - \overline{m})^2)^{\frac{3}{2}}}$;
(2) *threshold* = $\overline{m} + \gamma(\sigma + skew)$;
(3) *rel* = $\emptyset$;
**for** every document in the testing set {
$relevence(d) = \sum_{t \in T} pr_\beta(t)\tau(t, d)$;
if $(relevance(d) \geq threshold)$
$rel = rel \cup \{d\}$;
}

The time complexity of Algorithm TF1F in the testing phase is $O(nm) + O(mqu) = O(m(n + qu)) = O(mqu)$ since it only needs a traversal through each incoming document, where $q$ is the average size of testing documents; $u$ is the size of the testing set, and usually $n < u$.

Based on the above analysis, we believe that both algorithms for the topic filtering stage are efficient.

## IV. Filtering Based On Pattern Mining

After the topic filtering task has been carried out, the most irrelevant documents have been removed from testing sets. The second stage is to process the remaining documents using pattern mining technologies.

### A. Pattern Taxonomy Mining

A sequential pattern $s =< t_1, \ldots, t_r > (t_i \subseteq T)$ is an ordered list of terms. A sequence $s_1 =< x_1, \ldots, x_i >$ is a sub-sequence of another sequence $s_2 =< y_1, \ldots, y_j >$, denoted by $s_1 \sqsubseteq s_2$, iff $\exists j_1, \ldots, j_y$ such that $1 \leq j_1 < j_2 \ldots < j_y \leq j$ and $x_1 = y_{j_1}, x_2 = y_{j_2}, \ldots, x_i = y_{j_y}$. Given a sequence database $D$ and a minimum support threshold $\delta$, the problem of sequential pattern mining is to find the complete set of sub-sequences whose support is greater than $\delta$ in $D$. Also, such sub-sequences are called ***frequent sequential patterns***.

Not all frequent patterns are useful. The shorter patterns with the same support values as its parent are considered redundant and meaningless patterns, and need to be eliminated. It is expected to keep only the larger patterns or the patterns with a larger support value. The closed pattern mining is aimed at eradicating these short and meaningless closed patterns.

For example, in Table III and IV, a document is split in paragraphs and each paragraph is tread as a transaction. So a given document $d_i$ yields a set of paragraphs $DP = \{dp_1, dp_2, \ldots, dp_6\}$, and duplicate terms removed. Let $min\_sup = 50\%$ giving rise to the ten frequent patterns of Table III. Table IV illustrates these frequent patterns and their covering sets. After prune the non-closed pattern, there are only three patterns are closed sequential patterns: $\{t_3, t_4, t_6\}$, $\{t_1, t_2\}$ and $\{t_6\}$. The closed sequential patterns are structured into a taxonomy by using the $is\_a$ (or "subset") relation.

TABLE III.
A SET OF PARAGRAPHS

| $Parapgraph$ | $Terms$ |
|---|---|
| $dp_1$ | $t_1$ $t_2$ |
| $dp_2$ | $t_3$ $t_4$ $t_6$ |
| $dp_3$ | $t_3$ $t_4$ $t_5$ $t_6$ |
| $dp_4$ | $t_3$ $t_4$ $t_5$ $t_6$ |
| $dp_5$ | $t_1$ $t_2$ $t_6$ $t_7$ |
| $dp_6$ | $t_1$ $t_2$ $t_6$ $t_7$ |

TABLE IV.
FREQUENT PATTERNS AND COVERING SETS

| $Frequent\ Pattern$ | $Covering\ Set$ |
|---|---|
| $\{\mathbf{t_3}, \mathbf{t_4}, \mathbf{t_6}\}$ | $\{dp_2, dp_3, dp_4\}$ |
| $\{t_3, t_4\}$ | $\{dp_2, dp_3, dp_4\}$ |
| $\{t_3, t_6\}$ | $\{dp_2, dp_3, dp_4\}$ |
| $\{t_4, t_6\}$ | $\{dp_2, dp_3, dp_4\}$ |
| $\{t_3\}$ | $\{dp_2, dp_3, dp_4\}$ |
| $\{t_4\}$ | $\{dp_2, dp_3, dp_4\}$ |
| $\{\mathbf{t_1}, \mathbf{t_2}\}$ | $\{dp_1, dp_5, dp_6\}$ |
| $\{t_1\}$ | $\{dp_1, dp_5, dp_6\}$ |
| $\{t_2\}$ | $\{dp_1, dp_5, dp_6\}$ |
| $\{\mathbf{t_6}\}$ | $\{dp_2, dp_3, dp_4, dp_5, dp_6\}$ |

After the taxonomy is constructed, we first need to evaluate a term's support and then calculate a specificity value for each pattern. The evaluation of term supports (weights) is different to the normal term-based approaches. In the term based approaches, a component of a given term's weighting is based on its appearance in documents. However, in a taxonomy, terms are weighted according to their appearance in discovered patterns.

Formally, for all positive document $d_i \in D^+$, we first deploy its closed patterns on a common set of terms $T$ in order to obtain the following r-patterns:

$$\overrightarrow{d_i} =< (t_{i_1}, n_{i_1}), (t_{i_2}, n_{i_2}), \ldots, (t_{i_m}, n_{i_m}) > \qquad (6)$$

where $t_{i_j}$ in pair $(t_{i_j}, n_{i_j})$ denotes a single term and $n_{i_j}$ is its *support* in $d_i$ which is the number of closed patterns that contain $t_{i_j}$.

These r-patterns are composed by using an association mapping (see Eq. 2), and then the supports of the terms in $D^+$ are calculated.

### B. Document Ranking Algorithms

An algorithm, $SPMining$, was proposed in [20] to find all closed sequential patterns. For every positive document, the $SPMining$ algorithm is first called giving rise to a set of closed sequential patterns $SP$. Then, all discovered patterns in a positive document are composed into an r-pattern giving rise to a set of r-patterns $RP$. Thereafter, the support is calculated for all terms that appear in the r-patterns, where the normal forms $\beta(p)$ (see Eq. 1) for all r-pattern $p \in RP$ is used.

Algorithm PTM2 describes the training process of pattern taxonomy mining. For every positive document, the SPMining algorithm is first called in step (2) giving rise to a set of closed sequential patterns $SP$. Additionally, all

**Algorithm PTM2** ($D^+$, *min_sup*)
**Input:** $D^+$; minimum support, *min_sup*.
**Output:** a set of r-patterns $RP$, and supports of terms.
(1) $RP = \emptyset$;
(2) **for** (document $d \in D^+$) {
  let $DP$ be the set of paragraphes in $d$;
  //sequential pattern mining in a set of paragraphes
  $SP$ = SPMining(*DP*, *min_sup*);
  $\overrightarrow{d} = \emptyset$;
  **for** (pattern $p_i \in SP$) {
    $p = \{(t, 1)|t \in p_i\}$;
    $\overrightarrow{d} = \overrightarrow{d} \oplus p$ }
  $RP = RP \cup \{\overrightarrow{d}\}$ }
(3) $T = \{t|(t, w) \in p, p \in RP\}$;
  **for** (term $t \in T$) $support(t) = 0$;
(4) **for** (r-pattern $p \in RP$)
  **for** ($(t, w) \in \beta(p)$)
    $support(t) = support(t) + w$;

discovered patterns in a positive document are composed into an r-pattern giving rise to a set of r-patterns $RP$ in step (2). Thereafter in step (3), the support is calculated for all terms that appear in the r-patterns. Finally, in step (4), the normal forms $\beta(p)$ for all r-pattern $p \in RP$ is used.

After the support of terms have been computed from the training set, a given pattern's specificity to the given topic can be defined as follows:

$$spe(p) = \sum_{t \in p} support(t).$$

It is also easy to verify $spe(p_1) \leq spe(p_2)$ if $p_1$ is a sub-pattern of pattern $p_2$. This property shows that a document should be assigned a large weight if it contains large patterns. Based on this observation, we will assign the following weight to a document $d$ for ranking documents in the second stage:

$$weight(d) = \sum_{t \in T} support(t)\tau(t, d).$$

## V. A Two-stage IF Model

This section illustrates how the Rough Set-based filtering approach is unified with a pattern-mining-based filtering model to develop a more robust and intelligent two-stage information filtering system.

### A. Objectives

The idea of integrating term-based approaches (topic filtering) and pattern-based approaches (pattern taxonomy mining) for IF systems has evolved from these two well established, but largely disparate fields. The main objectives of this research are in exploiting the advantages of term-based approaches and pattern-based approaches (data mining) within the one system.

The proposed two-stage IF system uses the strategies used in both batch filtering and routing filtering. In the first stage, a topic filtering method based on the Rough Set decision rules is used to develop an optimal threshold. All the unlikely relevant documents are filtered out. The remaining documents of the incoming stream will pass into the second stage. The pattern mining method at the second stage will work on only a relatively small amount of documents. The remaining documents are potentially with a higher relevance at the second stage. Thus, better ranking accuracy will yield in the routing filtering process.

### B. Advantages of T-SM

The first stage is recall-oriented, and aims to address the information overload issue. The objective of the second stage is to apply pattern mining techniques to rationalise the data relevance of the reduced document set after the first stage. This stage is precision-oriented, and it is leaning more toward solving the information mismatch problem. Overall, the decision making is completed in two steps. It balances the recall and precision of the system, thus improving the performance of the system.

The other advantage of the two-stage approach is that the user profiles have been used from two different angles to analyse the incoming data stream without adding or even reducing the computational intensity. In comparison, pattern mining is more time consuming. With topic filtering reducing the documents that need to go through the pattern mining, the two-stage model can be faster than the one stage pattern mining process. The two-stage model uses the user profiles twice with different profile-learning methods. The users' information needs are better understood throughout the overall filtering procedure.

## VI. Discussion

In the pattern taxonomy mining, a small $min\_sup$ is used to find interesting patterns because of patterns having a low frequency of occurrence. The consequence is that some noisy terms and their combinations (patterns) are also retained and that makes some negative documents obtain large weights in the pattern mining model. The pattern taxonomy mining is sensitive to the data noise. To deal with this phenomenon, a two-stage theory was put forward.

In this research, only positive documents in the training set were used to formulate the user profiles. Using the rough threshold model, a positive document can be described as an "r-pattern" and a probability function can be generated to describes the features of the set of positive documents.

In theoretical perspective, any incoming document with a larger probability value than the minimum score of positive documents in the training set should be considered as possibly relevant. However, in real life, user profiles can be very uncertain. Using the minimum positive score as the threshold will allow too many irrelevant documents into the second stage. In certain cases, when the user profiles are most specific, using the minimum score as the threshold would be appropriate. However, in most cases, the user profiles are not well defined; therefore, the higher score should be used.

It is obvious that no any sort of two-stage model practises the significant performance. In the proposed two-stage model, the rough threshold model were used

to remove the majority of the irrelevant documents in the first stage. The goal of the first stage is to produce a relatively small set of mostly relevant documents as the input for the second stage, pattern taxonomy mining. Therefore, we conclude that the significant improvement is due mainly to the success in the removal of the noisy information by the topic filtering stage.

## VII. Conclusion

This paper illustrates a new model which integrates topic filtering and pattern taxonomy mining together to alleviate information overload and mismatch problems. The proposed method has been evaluated using the standard TREC routing framework with encouraging results.

Compared with the single BM25, SVM, PTM stage methods and other possible types of "two-stage" models, the results of experiments on RCV1 collection demonstrate that the performance of information filtering can be significantly improved by the proposed new model. The substantial improvement is mainly due to the rough threshold model applied to the topic filtering in the first stage and the "semantic" nature of patterns in the second stage. This research provides a promising methodology for developing effective filtering systems based on positive feedback information.

## Acknowledgment

We thank the reviewers for their detailed comments.

## References

[1] N. J. Belkin and W. B. Croft, "Information filtering and information retrieval: two sides of the same coin?" *Commun. ACM*, vol. 35, no. 12, pp. 29–38, 1992.

[2] J. Mostafa, W. Lam, and M. Palakal, "A multilevel approach to intelligent information filtering: model, system, and evaluation," *ACM Transactions on Information Systems*, vol. 15, no. 4, pp. 368–399, 1997.

[3] S. Robertson and D. A. Hull, "The trec9 filtering track final report," in *TREC-9*, 2000.

[4] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison Wesley, 1999.

[5] S. Robertson and I. Soboroff, "The trec 2002 filtering track report," in *TREC 2002*, 2002.

[6] Y. Li, C. Zhang, and J. R. Swan, "An information filtering model on the web and its application in jobagent," *Knowledge-based Systems*, vol. 13, no. 5, pp. 285–296, 2000.

[7] X. Zhou, Y. Li, P. Bruza, S.-T. Wu, Y. Xu, and R. Y. K. Lau, "Using information filtering in web data mining process," in *Web Intelligence*, 2007, pp. 163–169.

[8] T. Qin, X.-D. Zhang, D.-S. Wang, T.-Y. Liu, W. Lai, and H. Li, "Ranking with multiple hyperplanes," in *SIGIR*, 2007, pp. 279–286.

[9] N. Fuhr, "Probabilistic models in information retrieval," *The Computer Journal*, vol. 35, no. 3, pp. 243–255, 1992.

[10] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proceedings of the 20th International Conference on Very Large Data Bases*, 1994, pp. 478–499.

[11] X. Yan, J. Han, and R. Afshar, "Clospan: mining closed sequential patterns in large datasets," in *Proceedings of SIAM Int. Conf. on Data Mining (SDM 03)*, San Francisco, USA, 2003, pp. 166–177.

[12] J. Han and K.-C. Chang, "Data mining for web intelligence," *Computer*, vol. 35, no. 11, pp. 64–70, 2002, tY - JOUR Han, Jiawei and Chang, K.C.-C.

[13] M. Zaki, "Spade: An efficient algorithm for mining frequent sequences," *Machine Learning*, vol. 40, pp. 31–60, 2001.

[14] M. Seno and G. Karypis, "Slpminer: An algorithm for finding frequent sequential patterns using length-decreasing support constraint," in *Proceedings of IEEE 2002 Int. Conf. on Data Mining (ICDM'02)*, 2002, pp. 418–425.

[15] Y. Huang and S. Lin, "Mining sequential patterns using graph search techniques," in *Proceedings of the 27th Annual Int. Computer Software and Applications Conf. (COMPSAC'03)*, Dallas, USA, 2003, pp. 4–9.

[16] H. Ahonen, O. Heinonen, M. Klemettinen, and A. I. Verkamo, "Applying data mining techniques for descriptive phrase extraction in digital document collections," in *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (ADL'98)*, Santa Barbara, CA, USA, 1998, pp. 2–11.

[17] S.-T. Wu, Y. Li, Y. Xu, B. Pham, and P. Chen, "Automatic pattern-taxonomy extraction for web mining," in *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence (WI 2004)*, Beijing, China, 2004, pp. 242–248.

[18] Y. Xu and Y. Li, "Generating concise association rules," in *CIKM*, 2007, pp. 781–790.

[19] N. Jindal and B. Liu, "Identifying comparative sentences in text documents," in *SIGIR*, 2006, pp. 244–251.

[20] S.-T. Wu, Y. Li, and Y. Xu, "Deploying approaches for pattern refinement in text mining," in *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006)*, 2006, pp. 1157–1161.

[21] Y. Li and N. Zhong, "Mining ontology for automatically acquiring web user information needs," *IEEE transactions on Knowledge and Data Engineering*, vol. 18, no. 4, pp. 554–568, 2006.

[22] Z. Pawlak, "Rough sets," *International Journal of Computer and Information Sciences*, vol. 11, pp. 341–356, 1982.